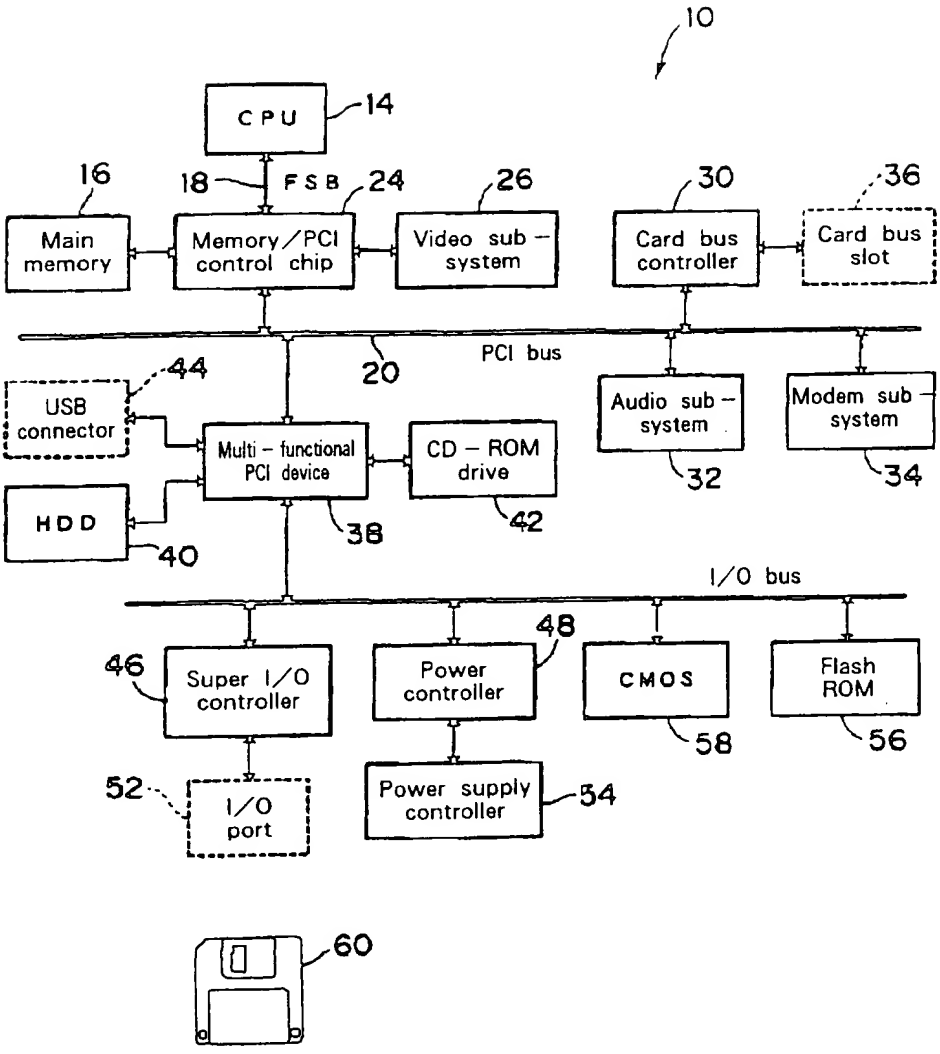
10
CPU
14
16
18
FSB
24
26
30
36
Main memory
Memory/PCI control chip
Video sub – system
Card bus controller
Card bus slot
PCI bus
20
44
USB connector
Multi – functional PCI device
CD – ROM drive
Audio sub – system
Modem sub – system
32
34
HDD
40
38
42
I/O bus
48
46
Super I/O controller
Power controller
CMOS
Flash ROM
58
56
52
I/O port
Power supply controller
54
60

Fig. 1

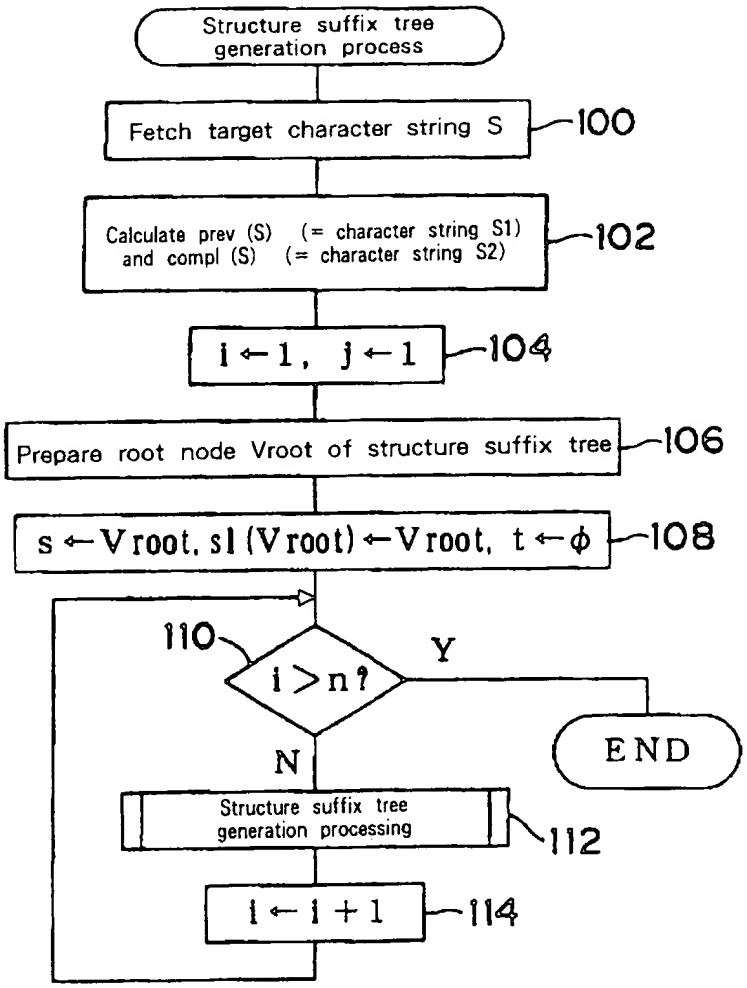
Structure suffix tree generation process
Fetch target character string S
100
Calculate prev (S) (= character string S1) and compl (S) (= character string S2)
102
i ← 1, j ← 1
104
Prepare root node Vroot of structure suffix tree
106
s ← Vroot, sl (Vroot) ← Vroot, t ← φ
108
110
i > n ?
Y
END
N
Structure suffix tree generation processing
112
i ← i + 1
114

Fig. 2

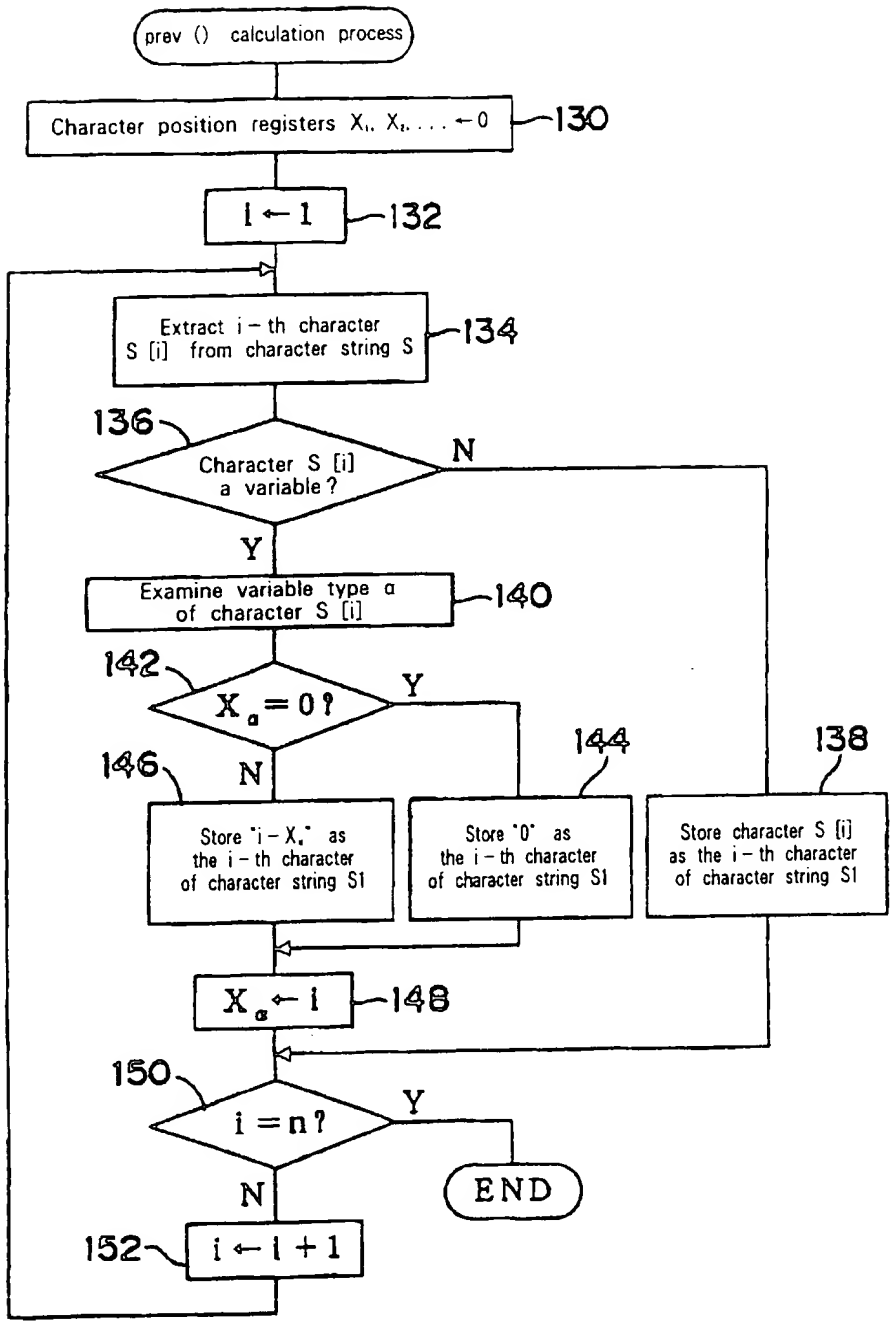
prev () calculation process
Character position registers X1, X2, . . . ← 0
130
i ← 1
132
Extract i – th character S [i] from character string S
134
136
Character S [i] a variable ?
N
Y
Examine variable type α of character S [i]
140
142
Xα = 0 ?
Y
144
138
146
N
Store "i – X1" as the i – th character of character string S1
Store "0" as the i – th character of character string S1
Store character S [i] as the i – th character of character string S1
Xα ← i
148
150
i = n ?
Y
END
N
152
i ← i + 1

Fig. 3

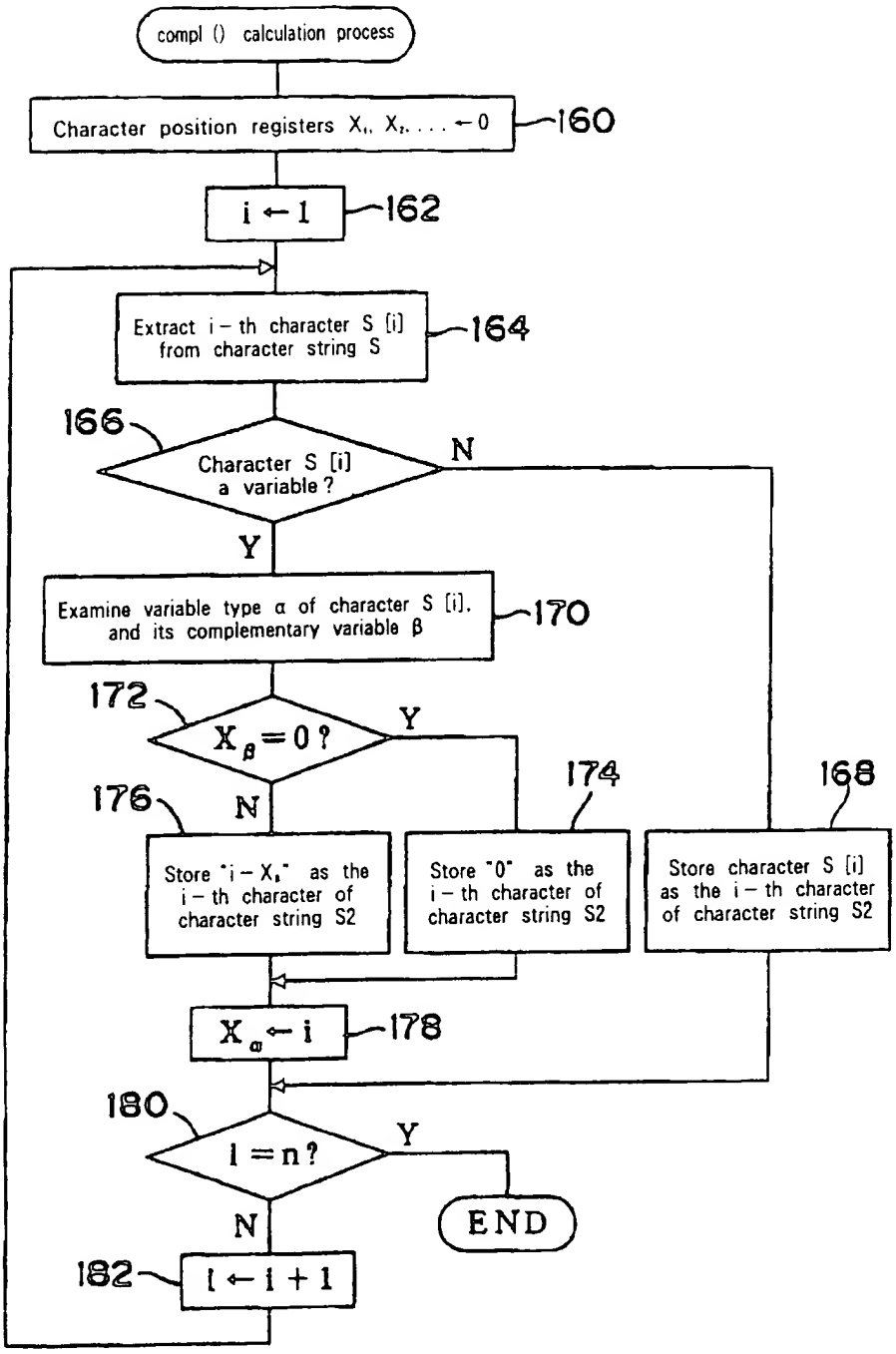
compl () calculation process
Character position registers X1, X2, . . . ← 0
160
i ← 1
162
Extract i – th character S [i] from character string S
164
166
Character S [i] a variable ?
N
Y
Examine variable type α of character S [i]. and its complementary variable β
170
172
Xβ = 0 ?
Y
N
174
168
176
Store "i – X1" as the i – th character of character string S2
Store "0" as the i – th character of character string S2
Store character S [i] as the i – th character of character string S2
Xα ← i
178
180
l = n ?
Y
N
END
182
l ← l + 1

Fig. 4

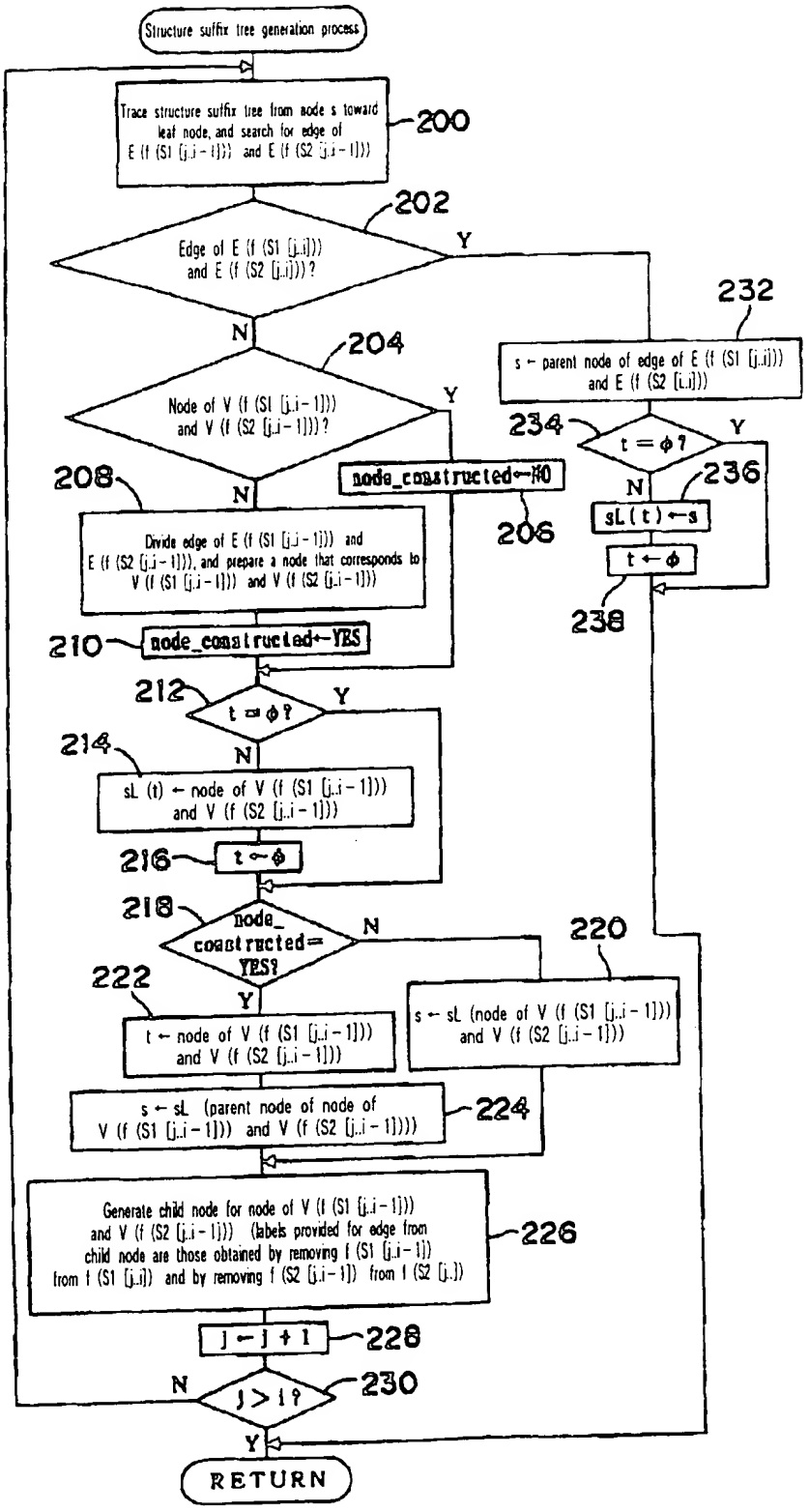
Structure suffix tree generation process
Trace structure suffix tree from node s toward leaf node, and search for edge of E (f (S1 [j..i-1])) and E (f (S2 [j..i-1]))
200
Edge of E (f (S1 [j..i])) and E (f (S2 [j..i]))?
202
Y
N
Node of V (f (S1 [j..i-1])) and V (f (S2 [j..i-1]))?
204
Y
N
node_constructed←NO
206
Divide edge of E (f (S1 [j..i-1])) and E (f (S2 [j..i-1])), and prepare a node that corresponds to V (f (S1 [j..i-1])) and V (f (S2 [j..i-1]))
208
node_constructed←YES
210
t = φ?
212
Y
N
sL (t) ← node of V (f (S1 [j..i-1])) and V (f (S2 [j..i-1]))
214
t ← φ
216
node_constructed= YES?
218
N
Y
t ← node of V (f (S1 [j..i-1])) and V (f (S2 [j..i-1]))
222
s ← sL (node of V (f (S1 [j..i-1])) and V (f (S2 [j..i-1]))
220
s ← sL (parent node of node of V (f (S1 [j..i-1])) and V (f (S2 [j..i-1])))
224
Generate child node for node of V (f (S1 [j..i-1])) and V (f (S2 [j..i-1])) (labels provided for edge from child node are those obtained by removing f (S1 [j..i-1]) from f (S1 [j..i]) and by removing f (S2 [j..i-1]) from f (S2 [j..i])
226
j ← j + 1
228
j > 1?
230
N
Y
RETURN
s ← parent node of edge of E (f (S1 [j..i])) and E (f (S2 [i..i]))
232
t = φ?
234
Y
N
sL (t) ← s
236
t ← φ
238

Fig. 5

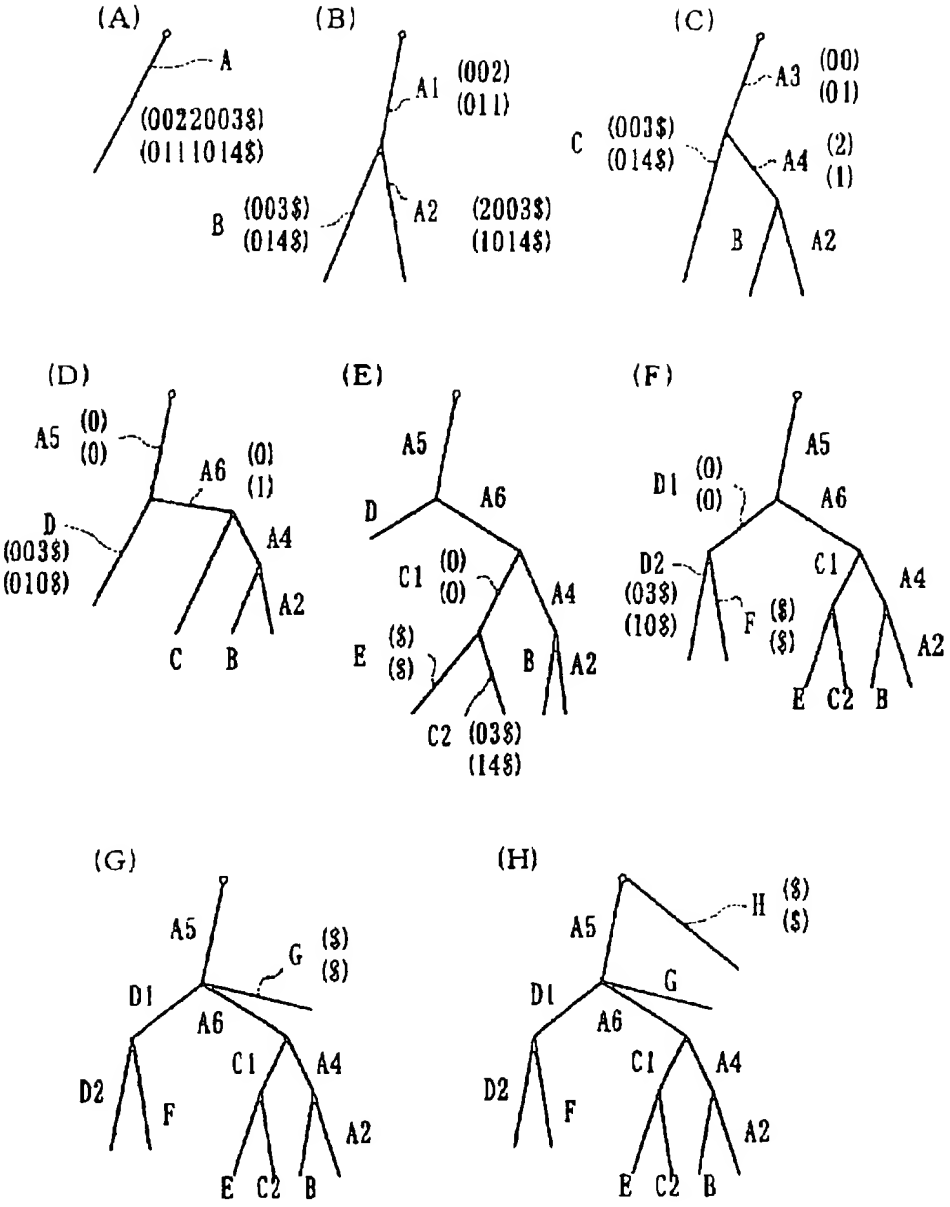
(A)
A
(00220038)
(01110148)
(B)
A1
(002)
(011)
B
(0038)
(0148)
A2
(2003$)
(1014$)
(C)
A3
(00)
(01)
C
(0038)
(0148)
A4
(2)
(1)
B
A2
(D)
A5
(0)
(0)
A6
(0)
(1)
D
(0038)
(0108)
A4
A2
C
B
(E)
A5
D
A6
C1
(0)
(0)
A4
E
($)
($)
B
A2
C2
(038)
(148)
(F)
A5
D1
(0)
(0)
A6
D2
(038)
(108)
C1
A4
F
($)
($)
A2
E
C2
B
(G)
A5
G
($)
($)
D1
A6
C1
A4
D2
F
A2
E
C2
B
(H)
A5
H
($)
($)
D1
G
A6
C1
A4
D2
F
A2
E
C2
B

Fig. 6

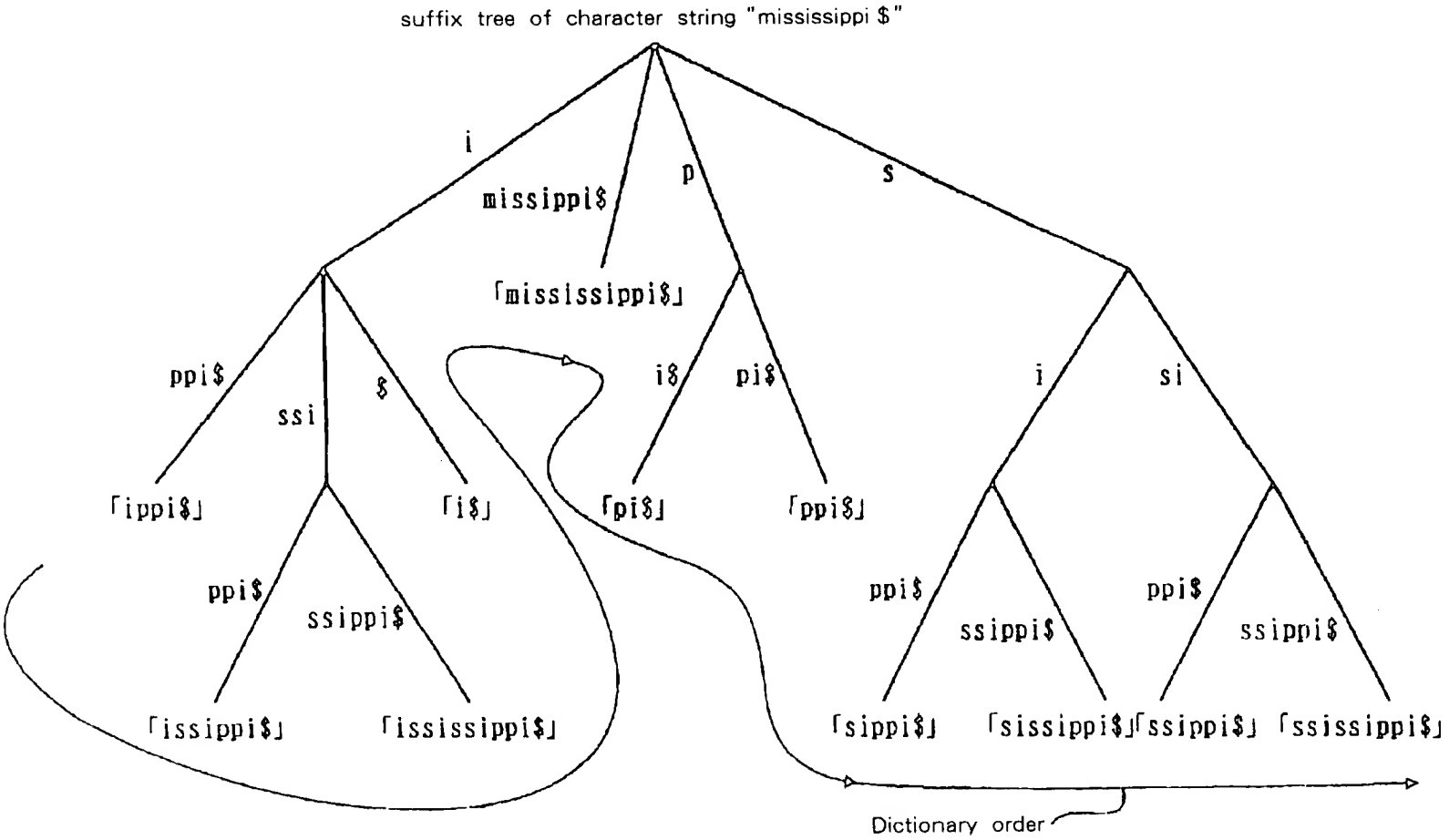
suffix tree of character string "mississippi $"
i
missippi$
p
s
「mississippi$」
ppi$
ssi
$
i$
pi$
i
si
「ippi$」
「i$」
「pi$」
「ppi$」
ppi$
ssippi$
ppi$
ssippi$
ppi$
ssippi$
「issippi$」
「ississippi$」
「sippi$」
「sissippi$」
「ssippi$」
「ssissippi$」
Dictionary order

Fig. 7